Biology Inspired techniques for Self Organization in dynamic Networks
IST-2001-38923

Ozalp Babaoglu

Dipartimento di Scienze dell'Informazione
Università di Bologna

Project funded by the Future and Emerging Technologies arm of the IST Programme

Motivation

- Current networked information systems are fragile (not robust), rigid (not adaptive) and are notoriously difficult to configure and maintain
- Many natural (biological, social) systems are exactly the opposite — they are *robust*, *adaptive* and *self-organizing* despite being highly *decentralized*
- Can we build information systems that are more “organic” or “life-like”?
- Do this by drawing inspiration from biology

Dynamic networks

- The problem is further aggravated in modern network structures
  - Mobile ad-hoc networks (MANET)
  - Overlay Networks
    - Peer-to-Peer systems
    - Grid computing
- Due to their extreme size and extreme dynamism

Project BISON

- Funded by IST-FET under FP5
- Partners
  - University of Bologna, Italy (Coordinator)
  - Telenor Communication AS, Norway
  - Technical University of Dresden, Germany
  - IDSIA, Lugano, Switzerland
- 1 January 2003 start date, duration 36 months
- Total cost €2,251,594
- EU funding €1,128,000
- URL: **`http://www.cs.unibo.it/bison`**

BISON objectives

- Complex adaptive system CAS are collections (swarm) of “agents”, acting in a decentralized and distributed fashion found in
  - nature and biological processes
  - social structures
  - economies, financial markets
- Behavior of CAS is often self-organizing, adaptive and robust (“nice properties”)
- We want to implement a number of functions on a variety of network structures using ideas from CAS
- Note that we are *not* interested in modelling or developing theories for explaining particular CAS

BISON expected results

- Decentralized, self-organizing, adaptive and robust solutions to important technological problems that arise in dynamic networks
- Systematic framework and a coherent set of heuristics to guide the synthesis of complex systems that solve interesting technological problems

BISON biological inspirations

- Social insects, ants
- Amoebae
- Chemotaxis
- Immune system
- Epidemics (gossip)
- Aggregation
- Neurons
- Regeneration

BISON biological inspirations

- Social insects, ants
- Amoebae
- Chemotaxis
- Immune system
- Epidemics (gossip)
- Aggregation
- Neurons
- Regeneration

BISON functions, services

- Routing (MANET)
- Power management (MANET)
- Load balancing
- Searching
- Collective computation
- Monitoring
- Topology management

BISON functions, services

- Routing (MANET)
- Power management (MANET)
- Load balancing
- Searching
- Collective computation
- Monitoring
- Topology management

Swarm intelligence

- The set of local agents that are equals (peers) forms the "swarm"
- The agents interact (locally)
- Each individual agent has very limited intelligence (ie, simple rules)
- But the *swarm* has a collective intelligence that can handle difficult challenges
- The intelligent behavior that the swarm exhibits (built from simple agents following simple rules) is called "emergence"

Emergence

- Emergence is all around us
  - a city
  - car traffic
  - the brain
  - the immune system
  - an ant colony
- *Emergent behavior* is collective behavior arising from the interaction of many autonomous units, where the units obey simple rules, and yet it is:
  - *complex* and *interesting* (maybe even *adaptive*)
  - *difficult to predict* from knowledge of the agents' rules

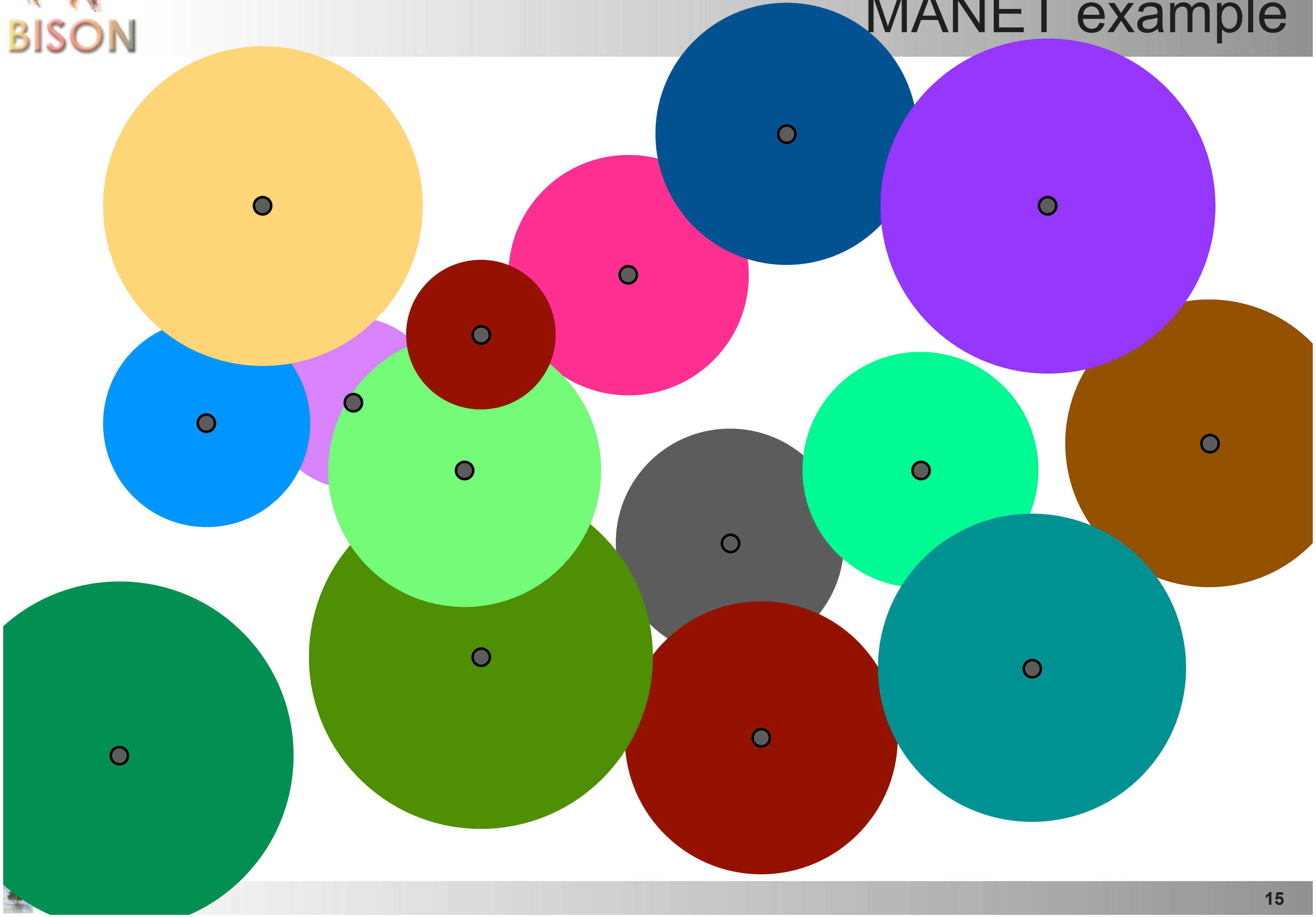
Back to BISON

- BISON applies these ideas to large-scale, dynamic networks of computers, PDAs, phones, etc. to solve important problems such as efficient routing of traffic; load balancing; search over distributed content; distributed computation

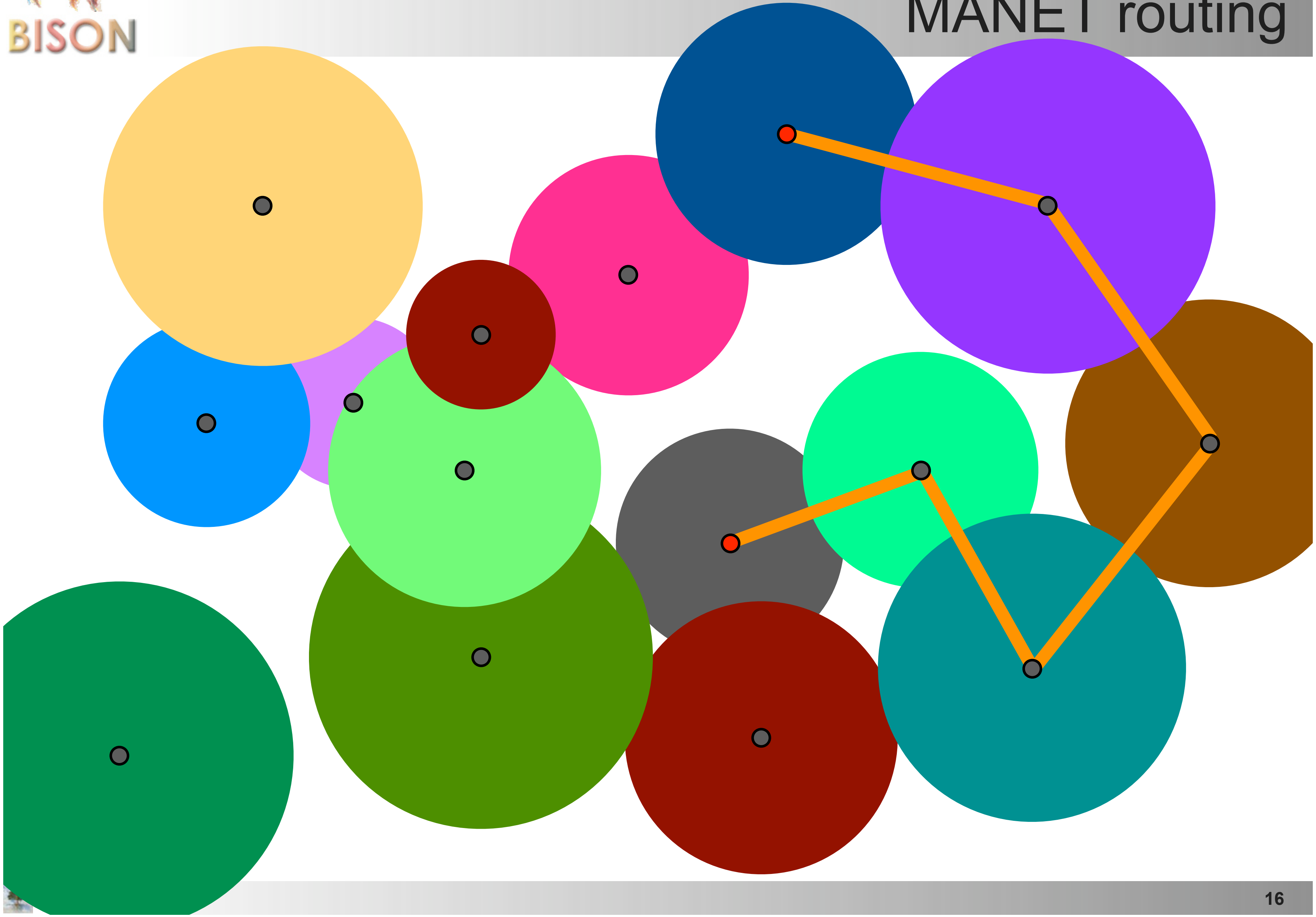
What is a MANET

- A network in which all communication is wireless:
  - One shared channel
  - Unreliable transmissions
  - Low bandwidth
- All nodes are mobile:
  - Nodes can enter and leave the network at any time
  - The topology changes constantly
- There is no fixed infrastructure:
  - All nodes act as terminals and routers
  - There is no central control or overview

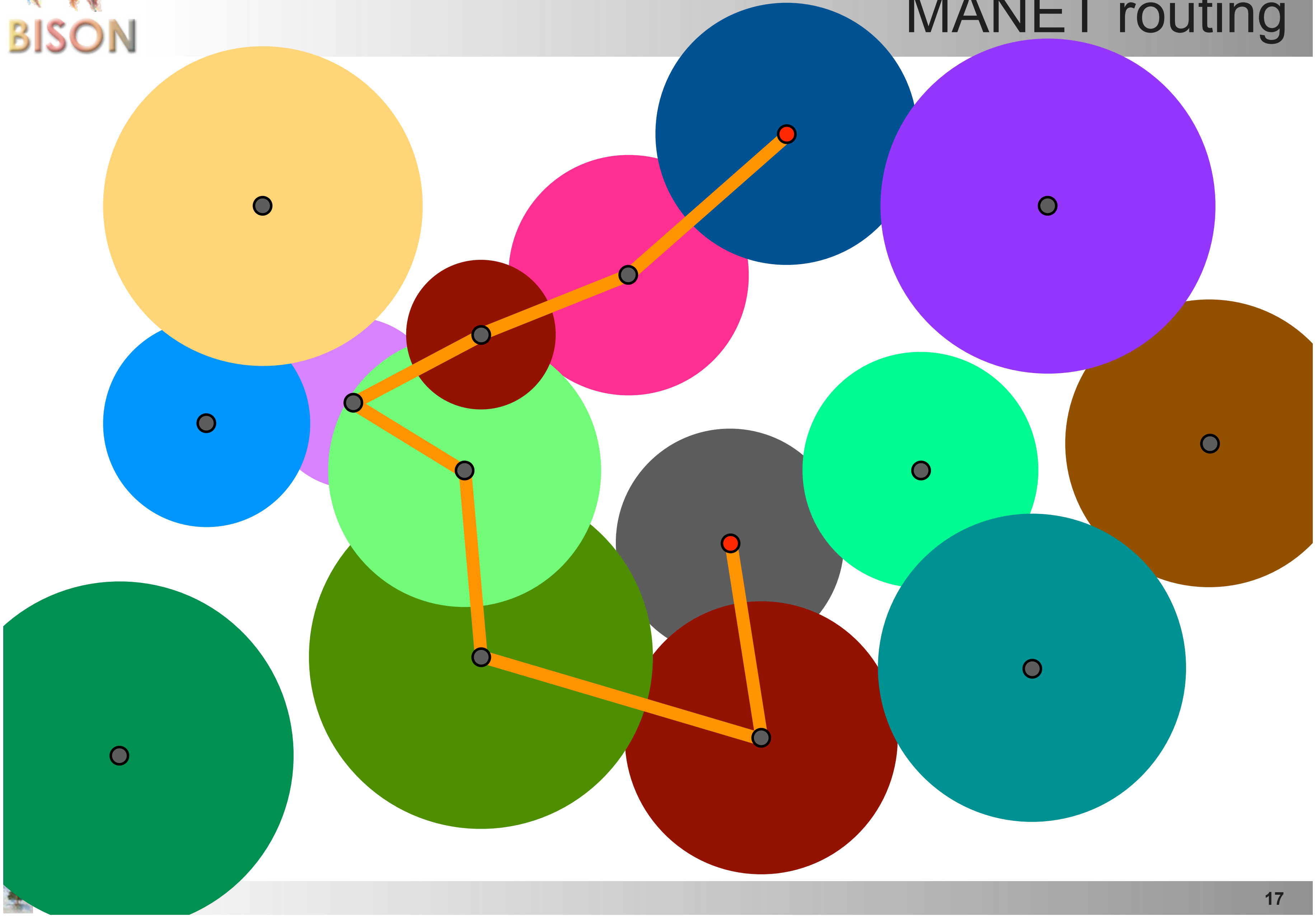
MANET example



MANET routing



MANET routing



MANET routing

- How to route traffic between nodes so that all those that can communicate do so and with (almost) minimal latency
- Despite the various sources of dynamism
  - Changing traffic patterns
  - Unreliable communication
  - Arrival-departure of nodes
  - Mobility of nodes
  - Changing radio coverage (battery drain)

Biological inspiration: Ant Colonies

- Foraging ant colonies can find shortest paths in a synergistic way in distributed and dynamic environments:
  - While moving back and forth between nest and food, ants mark their path by secreting pheromone
  - Step-by-step routing decisions are biased by the local intensity of pheromone field
  - Pheromone is the colony’s collective and distributed memory — it encodes the collectively learned quality of local routing choices toward destination target

Swarm routing with ants

- Desired emergent behavior: identification of good paths
- Simple rules:
  - explore the net, seeking a destination
  - lay a trace (pheromone) reflecting the quality of the path taken
  - traffic to be routed follows the traces with the strongest pheromone

How ants find food

First, they explore at random

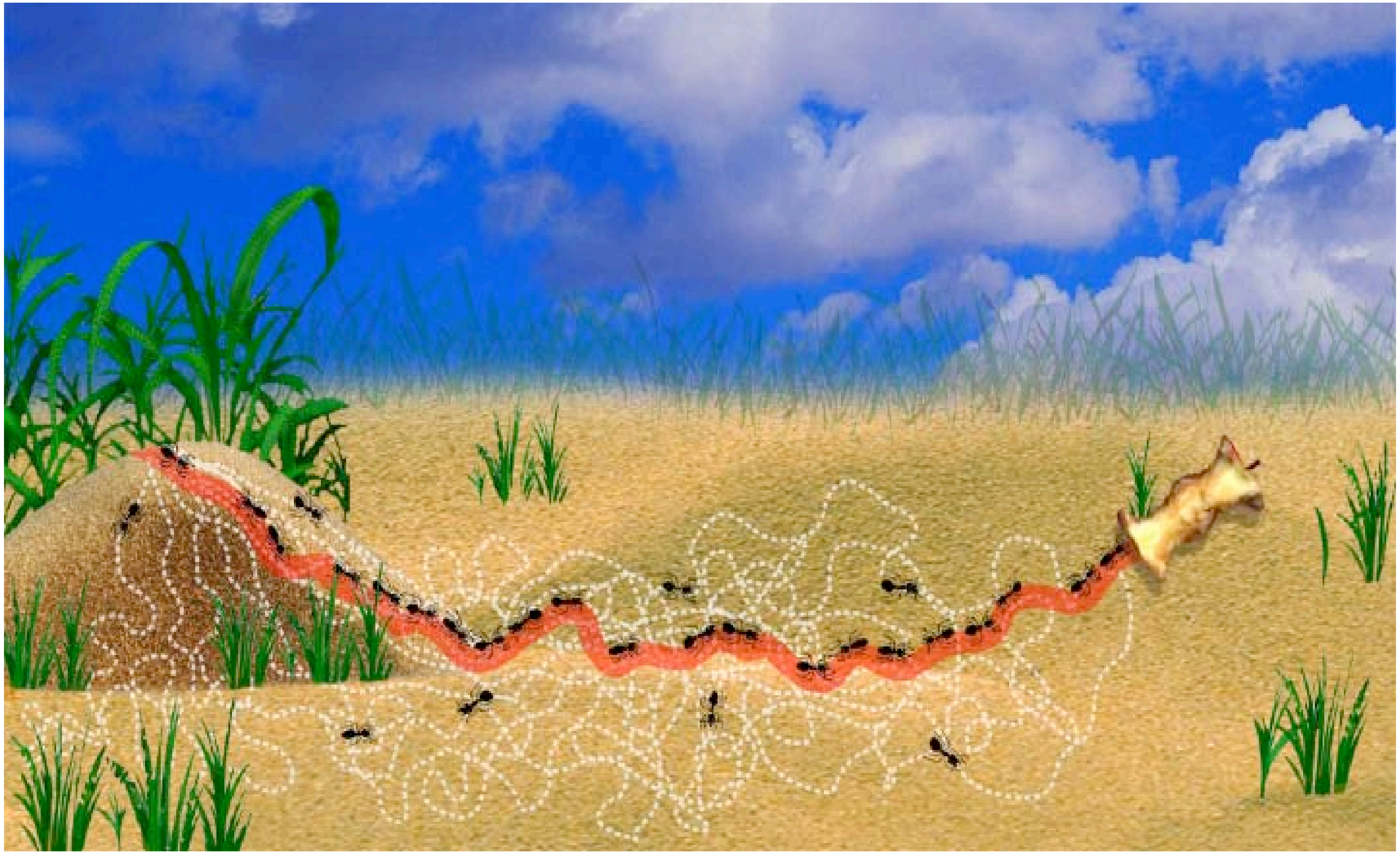
How ants find food

Individual ants mark their path by emitting a chemical substance — a pheromone — as they forage for food

How ants find food

Ants smell pheromone and they tend to choose path with strong pheromone concentration

How ants find food



Other ants use the pheromone to find the food source

How ants find food

When the “system” is interrupted, ants are able to adapt by rapidly adopting second best solutions

How ants find food

Social insects, following simple, individual rules, accomplish complex colony activities that are adaptive, robust and self-organizing

From ants to agents

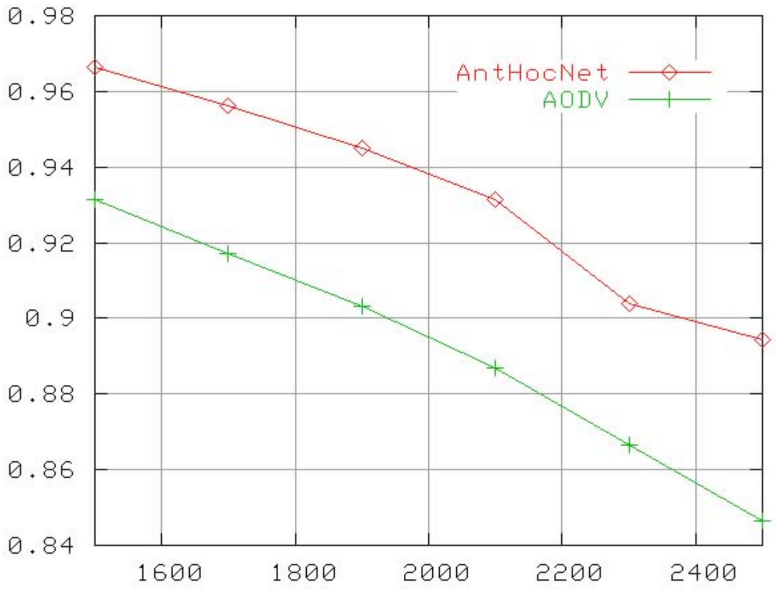
- Reverse-engineering of ant colony mechanisms leads to "Ant Colony Optimization"
  - Combinatorial optimization
  - Adaptive routing (AntNet)
- Multiple autonomous/concurrent agents (ants)
  - Solution constructed as a sequential decision process
  - Stochastic decision policy depending on pheromone
  - Use of solution outcomes to iteratively update pheromone

AntHocNet

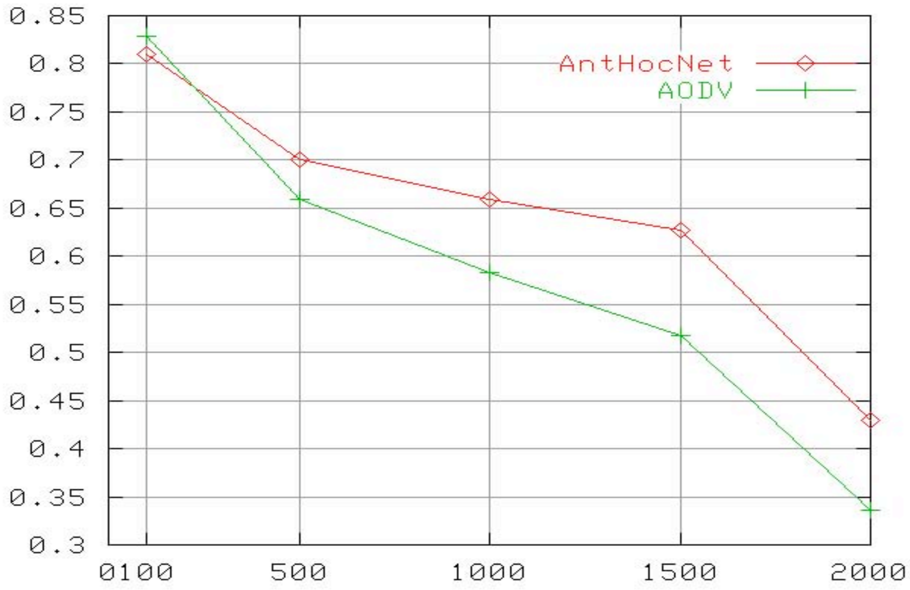
- Ant-based datagram routing:
  - Data are routed stochastically as datagrams according to pheromone tables
  - Pheromone entry: estimated quality of next hop choice for a destination
  - Ant agents sample paths and update pheromone tables
    - Reactive route setup
    - Proactive maintenance and updating
  - For each source-destination, a bundle of datagram paths is available (backup, multi-path spreading)

AntHocNet – Performance

Delivery ratio



Arena long edge

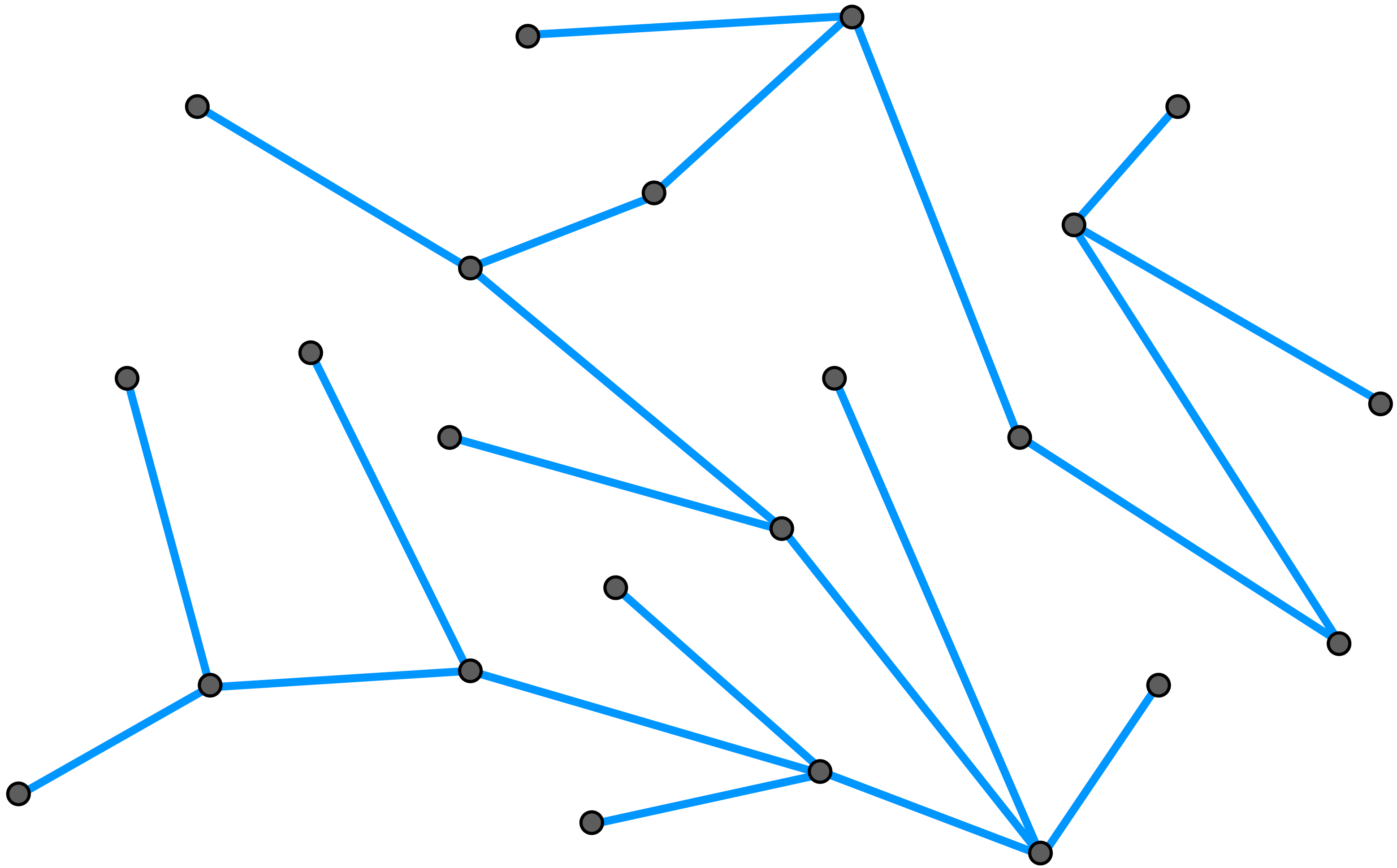


Number of nodes

Views, overlay networks

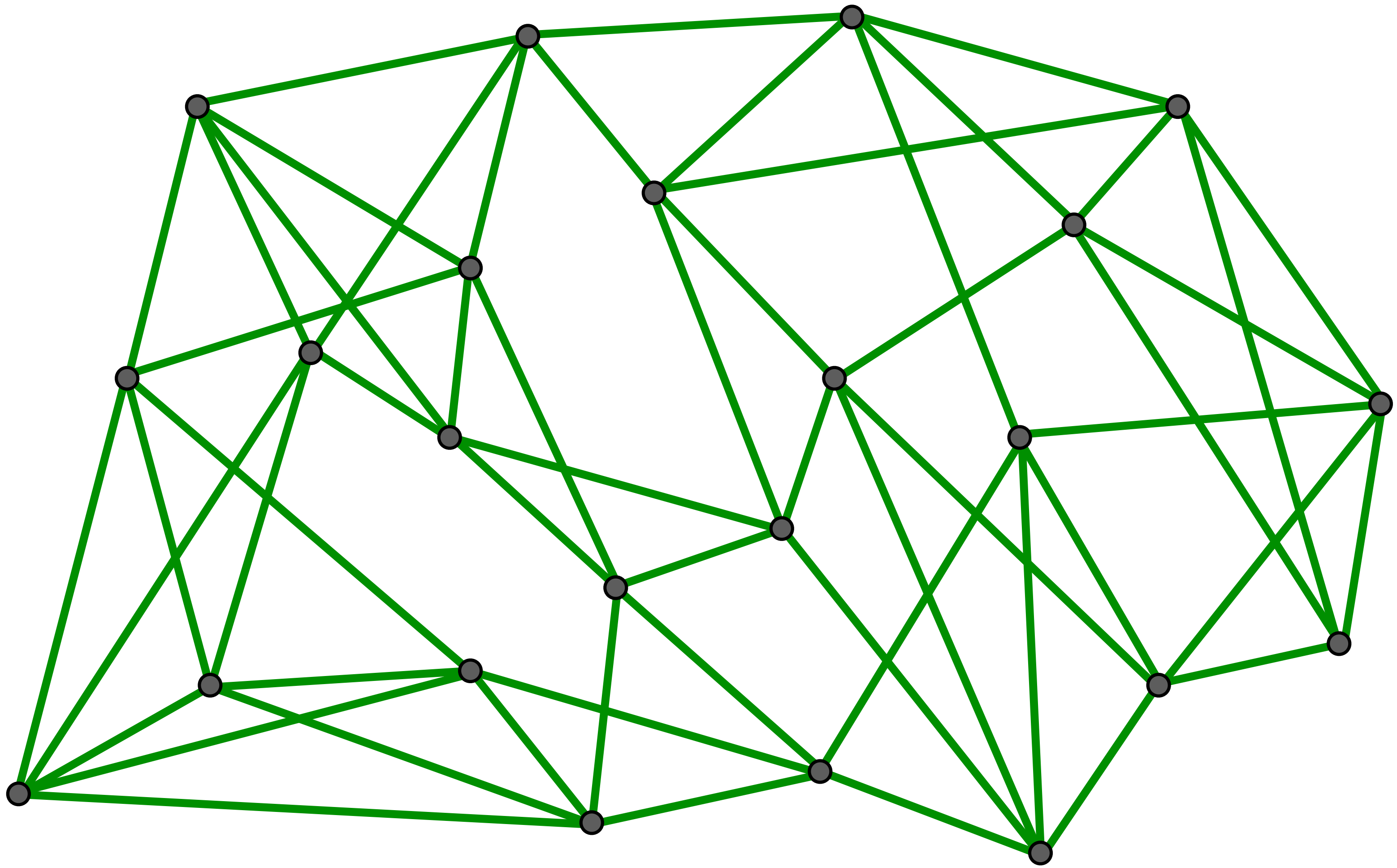
- The set of nodes that a peer knows about is called its view
- Typically, views are a (very) small subset of all nodes
- Views are typically dynamic since the set of nodes and the “knows” relation are highly dynamic (churn)
- Views define an *overlay network* with dynamic topology on top of the physical network

Overlay networks



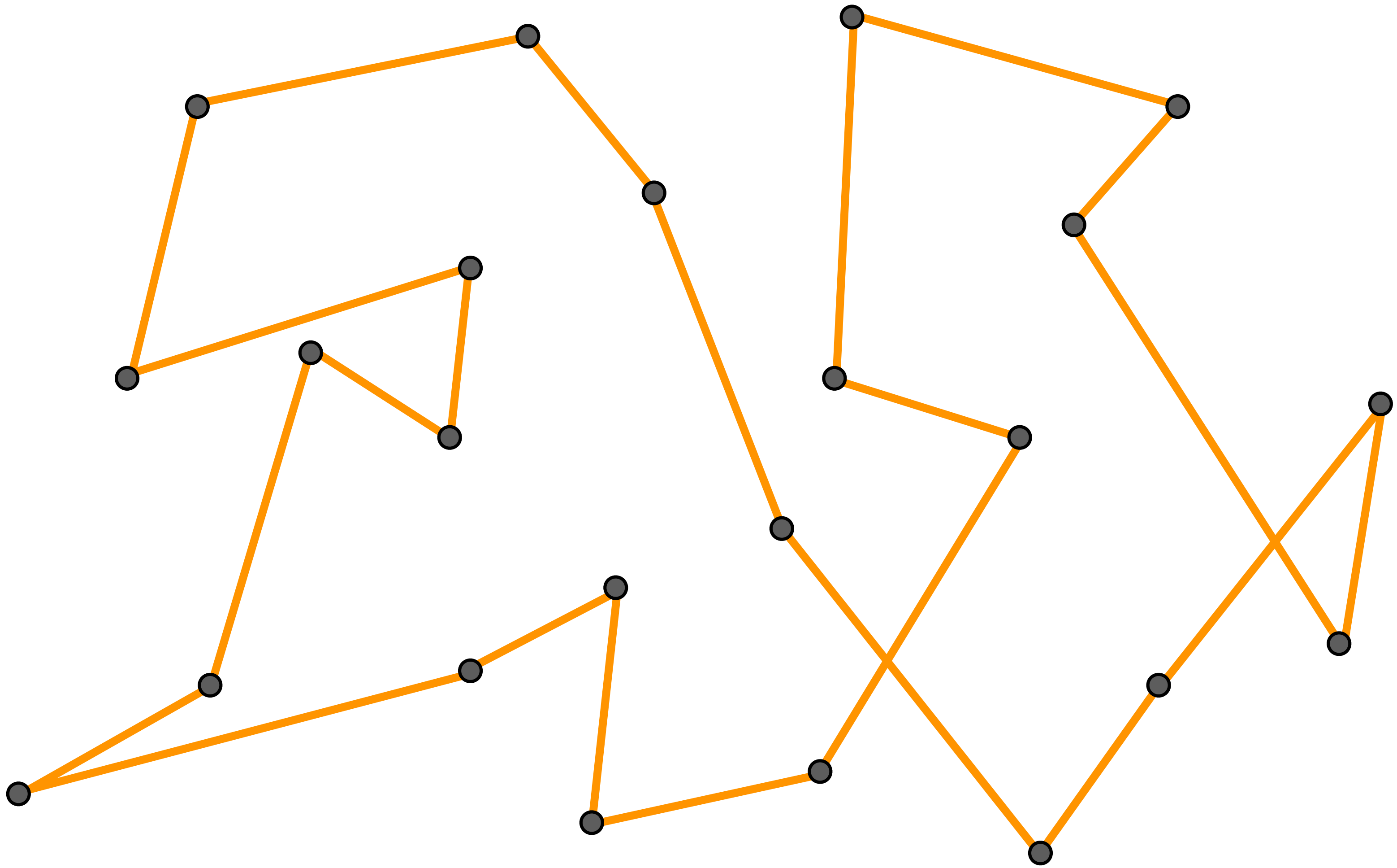
Physical network
"who has a communication link to whom"

Overlay networks



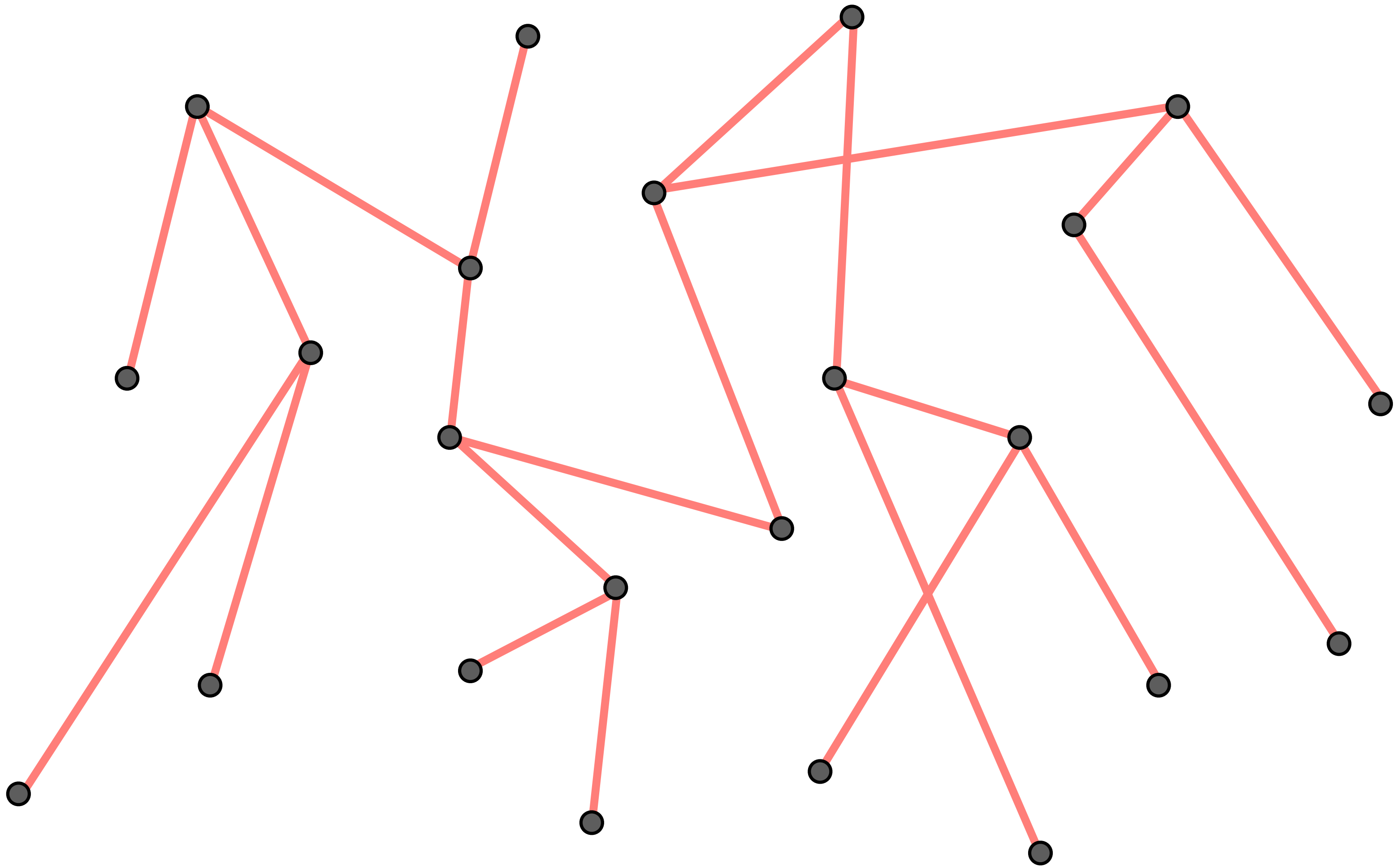
Logical network
"who can communicate with whom"

Overlay networks



Overlay network (ring)
"who knows whom"

Overlay networks



Overlay network (binary tree)
"who knows whom"

Functions over overlay networks

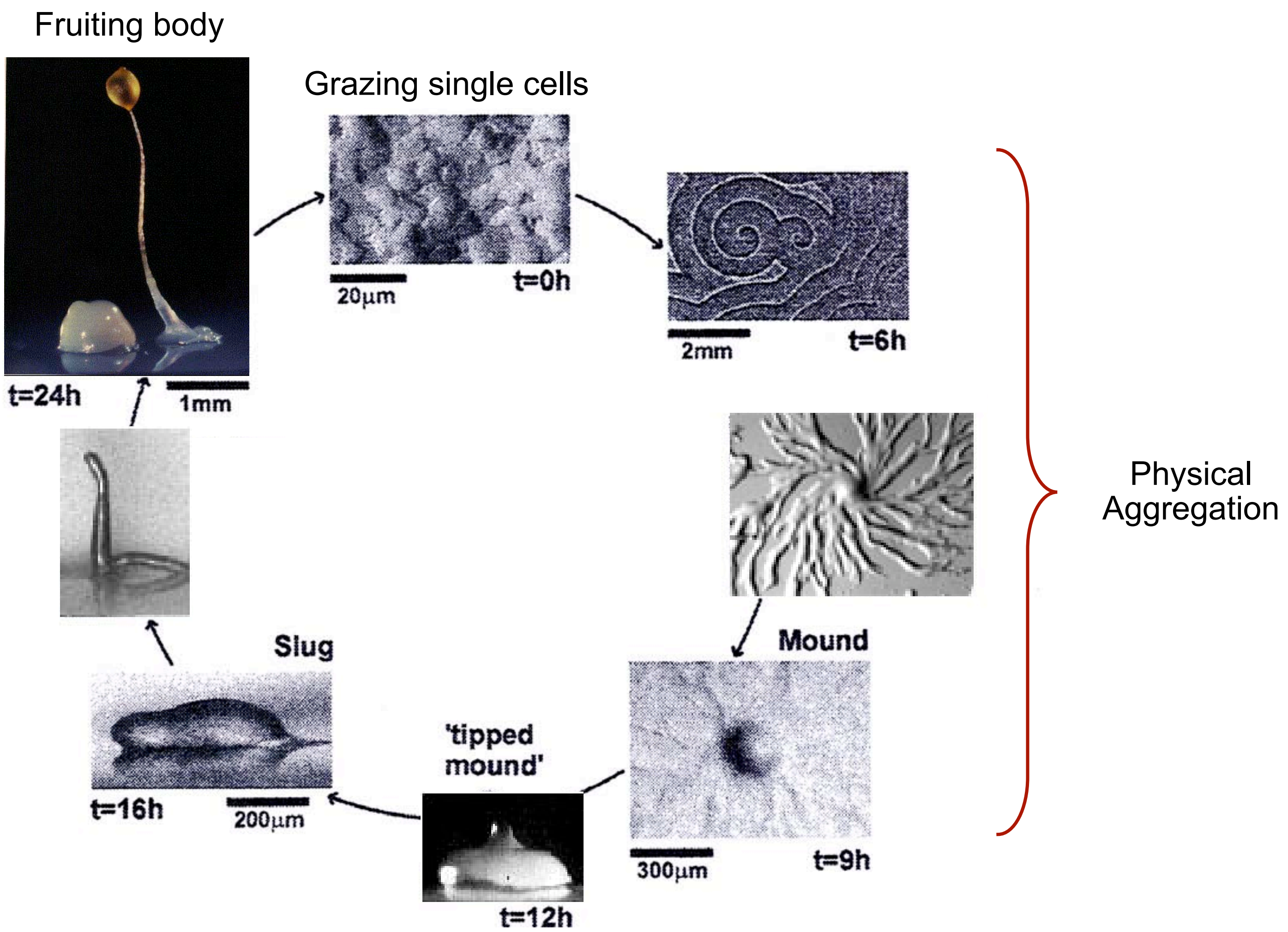
- *Load balancing*
  - Distribute “load” over the nodes as evenly as possible
- *Content searching*
  - Identify nodes that hold copies of certain “content”
- *Collective computation*
  - Compute arbitrary functions over local values
- *Topology management*
  - Build and maintain overlay networks with desired topologies

- Despite extreme size and dynamism (churn)

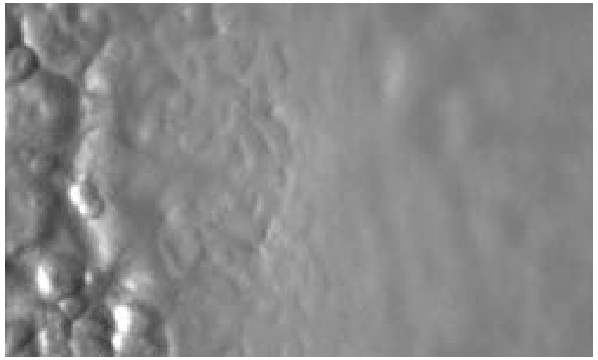
Load balancing
Biological inspiration: Dictyostelium

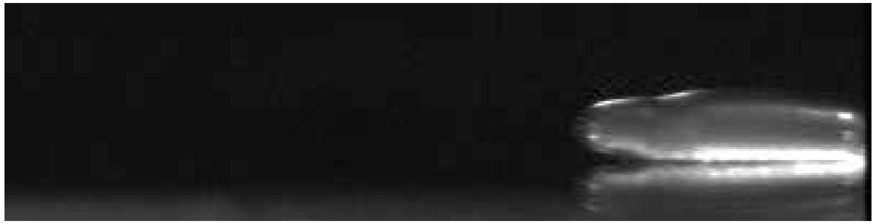
- *Dictyostelium* (slime mold) typically grows as separate, independent cells but interact to form a coordinated multicellular structure when induced by starvation
- Physical aggregation transforms multiple, independent cells into a single, multicellular organism
- The underlying mechanism is called *chemotaxis*: motion (taxis) along increasing gradients guided by diffusing chemical signals

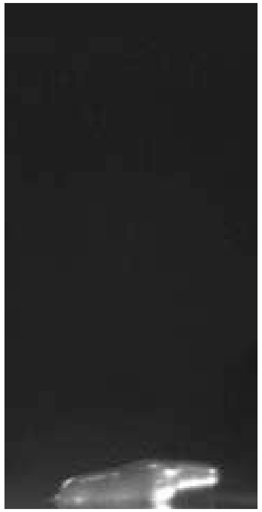
Life cycle of Dictyostelium



Life cycle of Dictyostelium



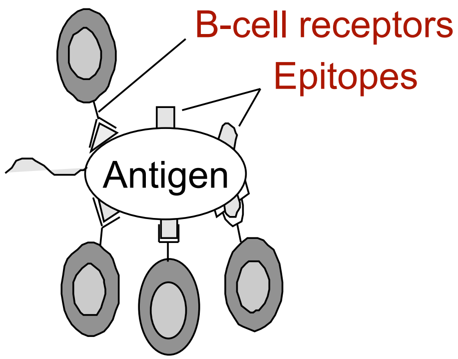




Load balancing

- Inverted chemotaxis that provokes motion along *decreasing* gradients and thus “anti-aggregate” — load balance
- Instance of a more general technique employing two mechanisms operating at different time scales:
  - rapidly diffusing “signal” (chemical substance, information)
  - slowly diffusing “matter” (cells, load)

Immune system basics

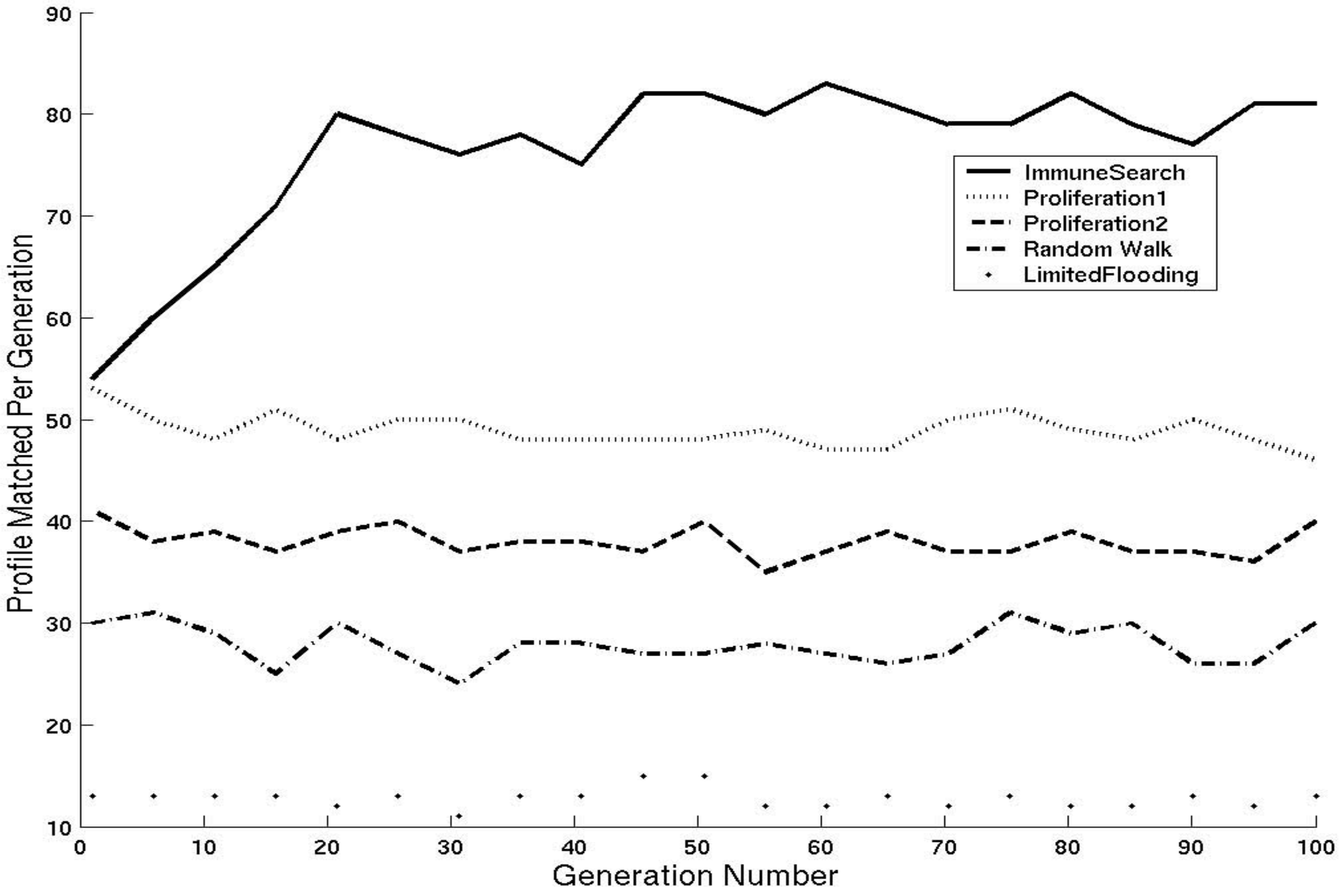


- Immune recognition is based on the complementary structure of the antibody (B-cell) receptors and a portion of the antigen called the *epitope*
- Antigen attack → Immune recognition → antibody activation, proliferation → antibody mutation (diversification)

ImmuneSearch

- Content (file) being searched for: “antigen”
- Queries: mobile “antibodies”
- Pattern recognition: affinity measure, similarity between content and query
- Search mechanisms:
  - **Proliferation**: antibodies replicate themselves when they find a good match
  - **Mutation**: antibodies undergo random transformations
  - **Clustering**: matching content is moved closer to the source of the query by rewiring the overlay network

Performance of ImmuneSearch



<u>Performance</u>: (proliferation+clustering) > proliferation > random walk > flooding

Collective computation (aggregation)

- Each node has a (numeric) local state
- Compute (global) aggregate function over the initial values at *all* nodes
- The aggregate value to be known (locally) at each node
- Examples of aggregate functions:
  - Average
  - Min-max
  - Geometric mean
  - Variance
  - Network size

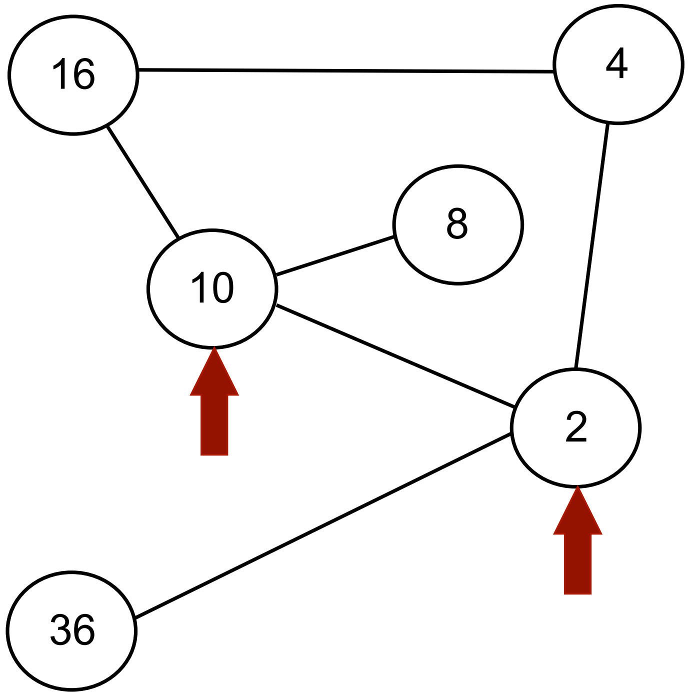
Biological inspiration: Viruses, epidemic spreading

- Each node periodically selects another (random) peer and exchanges local state information
- Each node updates its local state based on the information exchanged
- System fully symmetric — all nodes act identically
- Communication is symmetric — “push-pull” epidemic
- Proactive
- Many uses in distributed systems

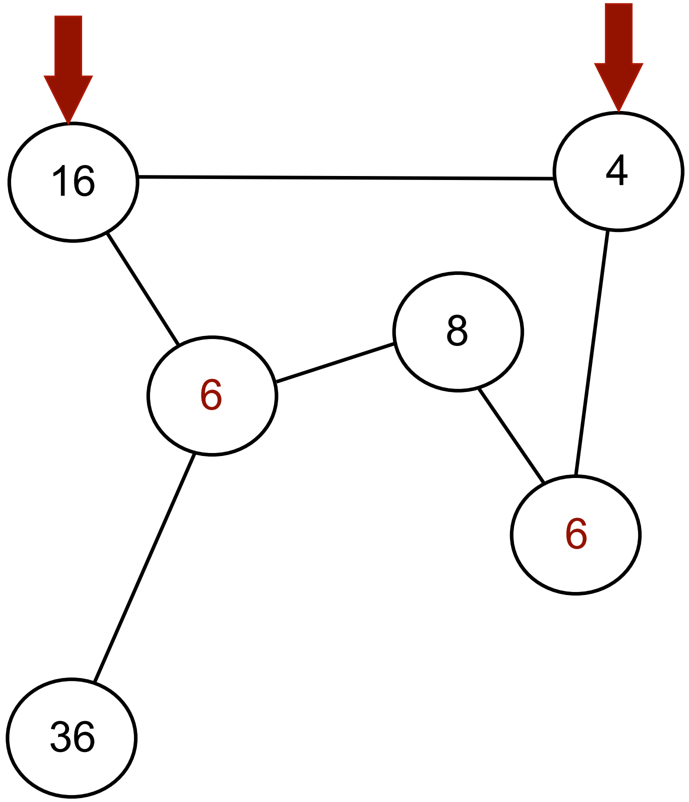
Aggregation through epidemics

- Local value S_p contains current estimate of the aggregate
- Suppose the (random) peer picked by node p is q
- Nodes p and q exchange current estimates
- Update local estimates
  - average: $S_p \leftarrow \frac{(S_p + S_q)}{2}$
  - geometric mean: $S_p \leftarrow \sqrt{(S_p S_q)}$
  - maximum: $S_p \leftarrow max(S_p, S_q)$
- Other, more complex functions built by combining elementary functions

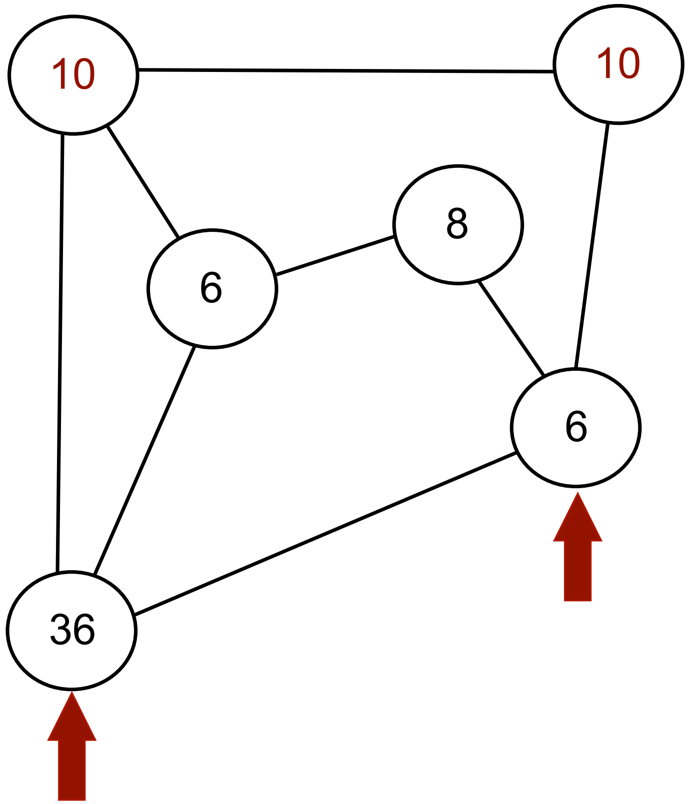
Aggregation example: averaging



Aggregation example: averaging



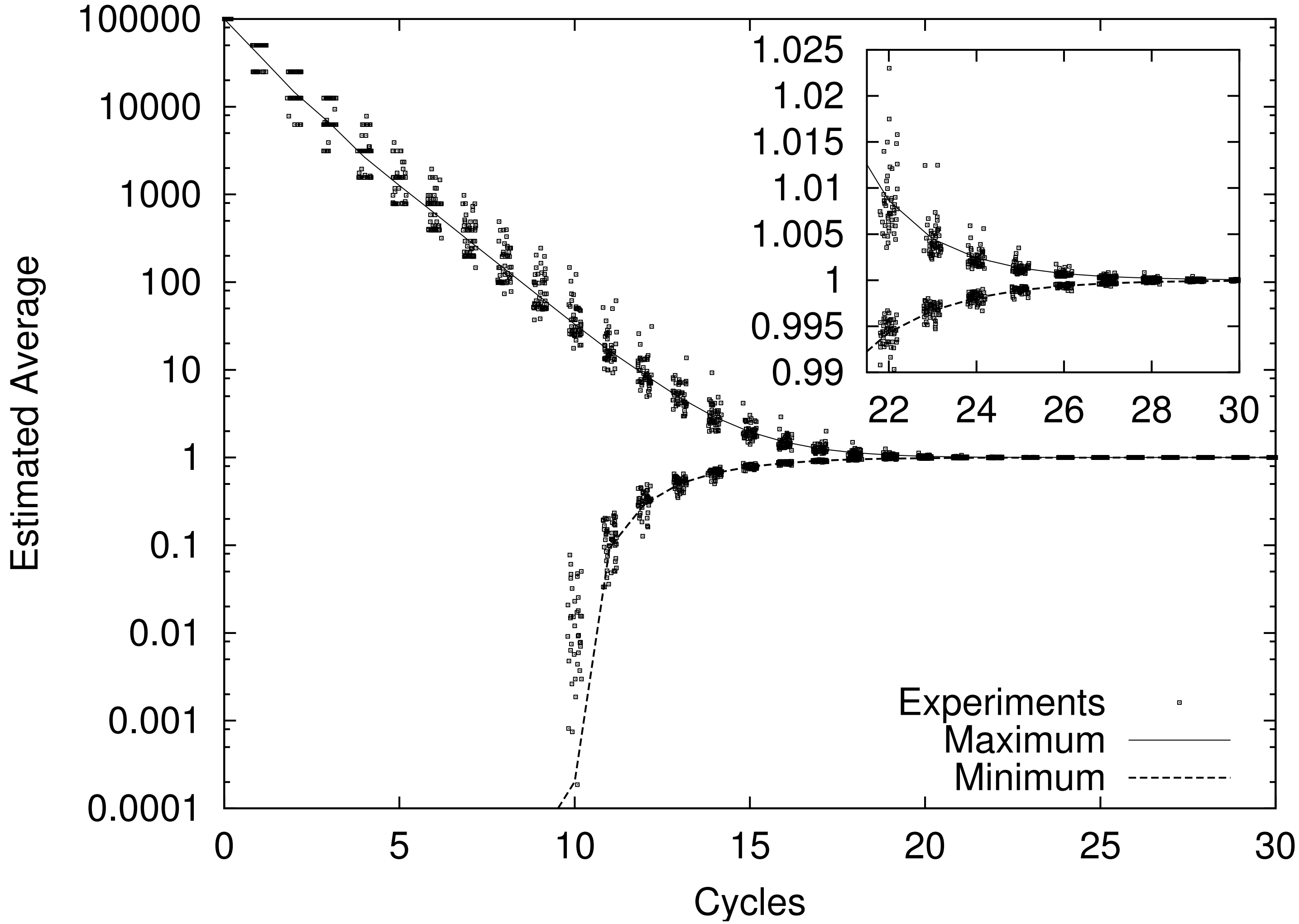
Aggregation example: averaging



Properties of epidemic-based aggregation

- In epidemic-based averaging, if the selected peer is a globally random sample, then the variance of the set of estimates decreases exponentially
- Extreme robustness to node and link failure and node dynamism (churn)

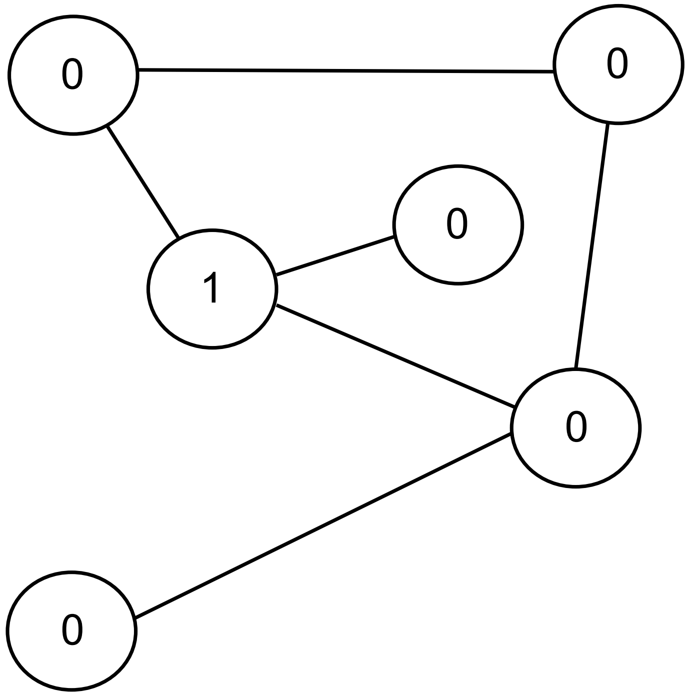
Exponential convergence of averaging



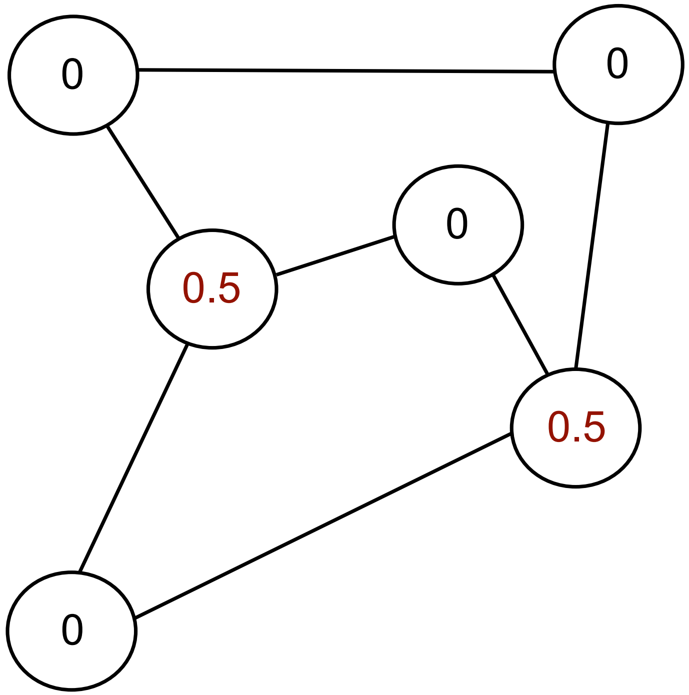
Network size estimation

- How to count the number of peers in an overlay network
  - Tens of million potential peers
  - Continual flux (churn)
  - Not allowed to “freeze” system
- Similar to conducting a census in a country the size of Italy without requiring people to stay at home on a given day

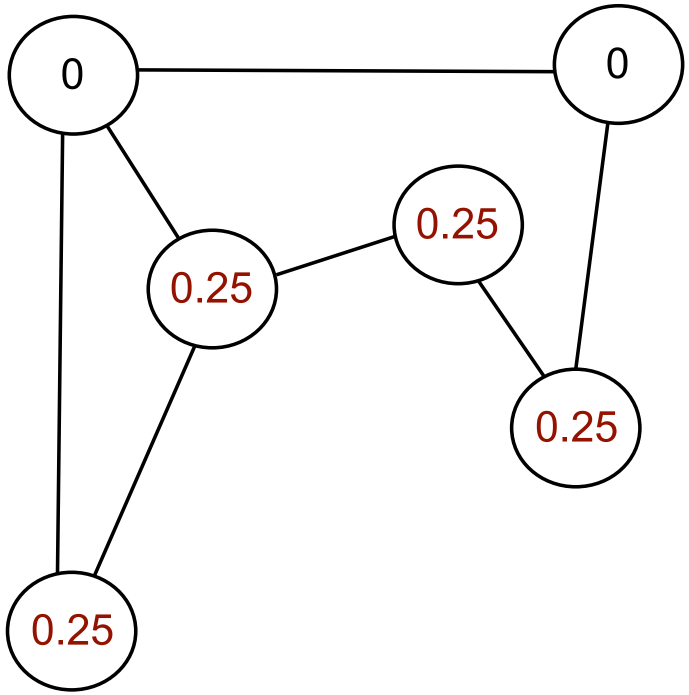
Network size estimation using averaging



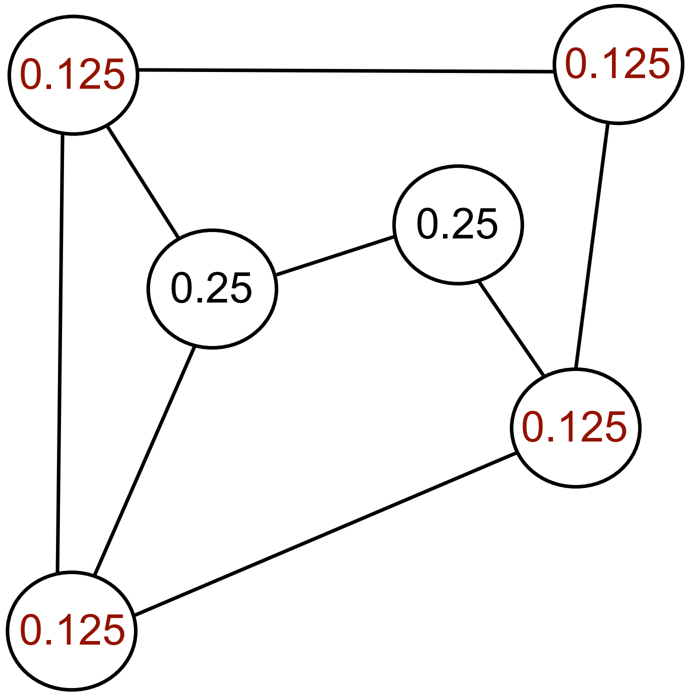
Network size estimation using averaging



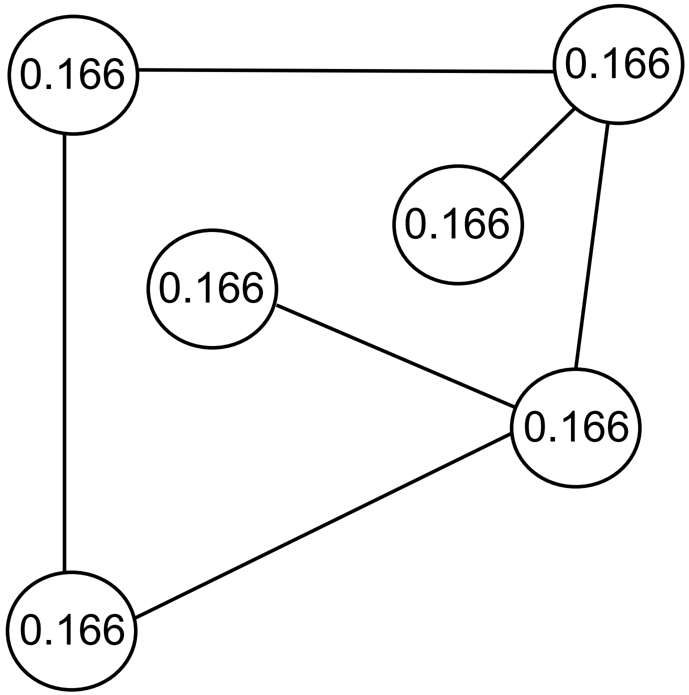
Network size estimation using averaging



Network size estimation using averaging

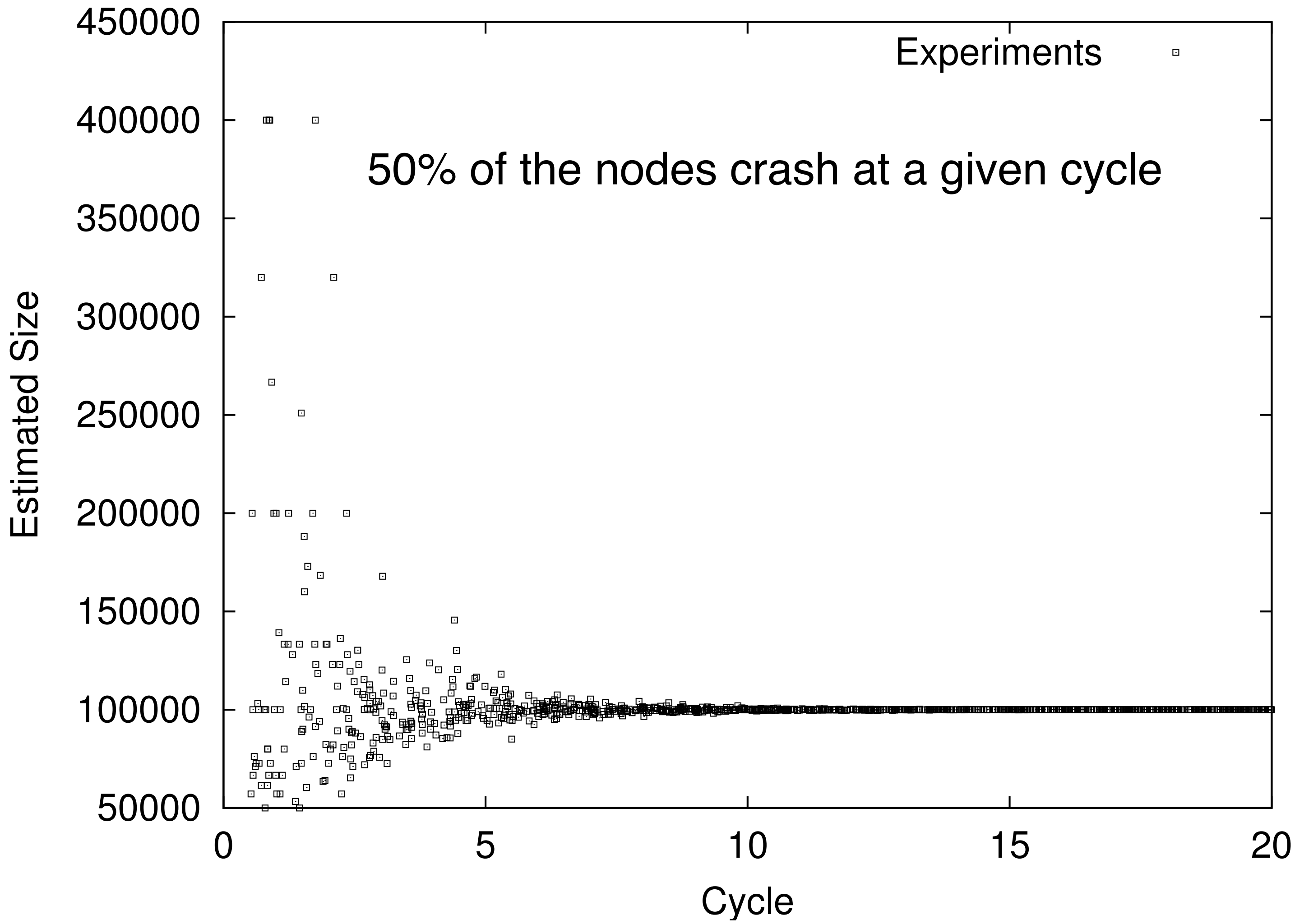


Network size estimation using averaging

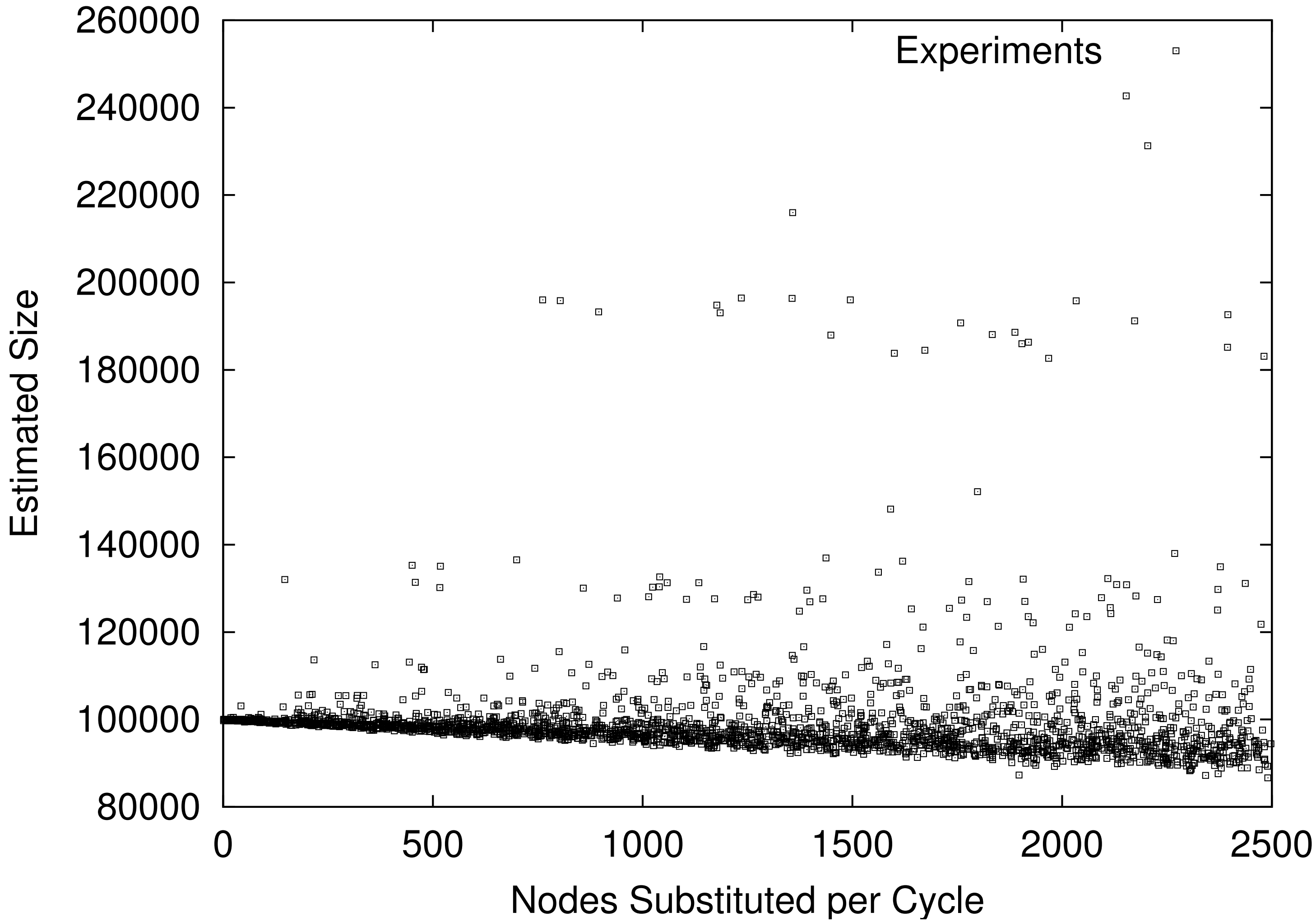


$1/0.166 = 6.02 \approx N$

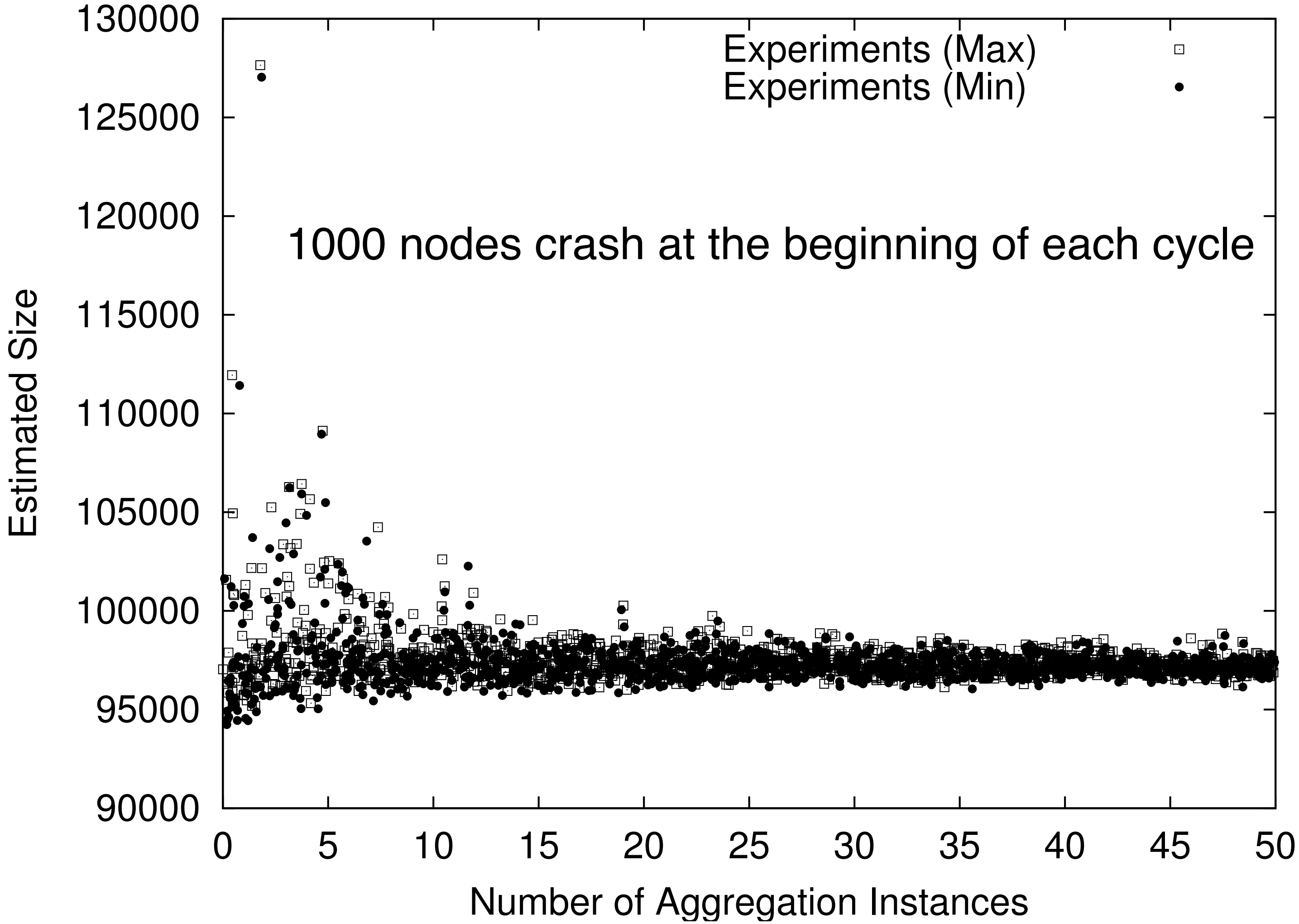
Network size estimation with node crashes



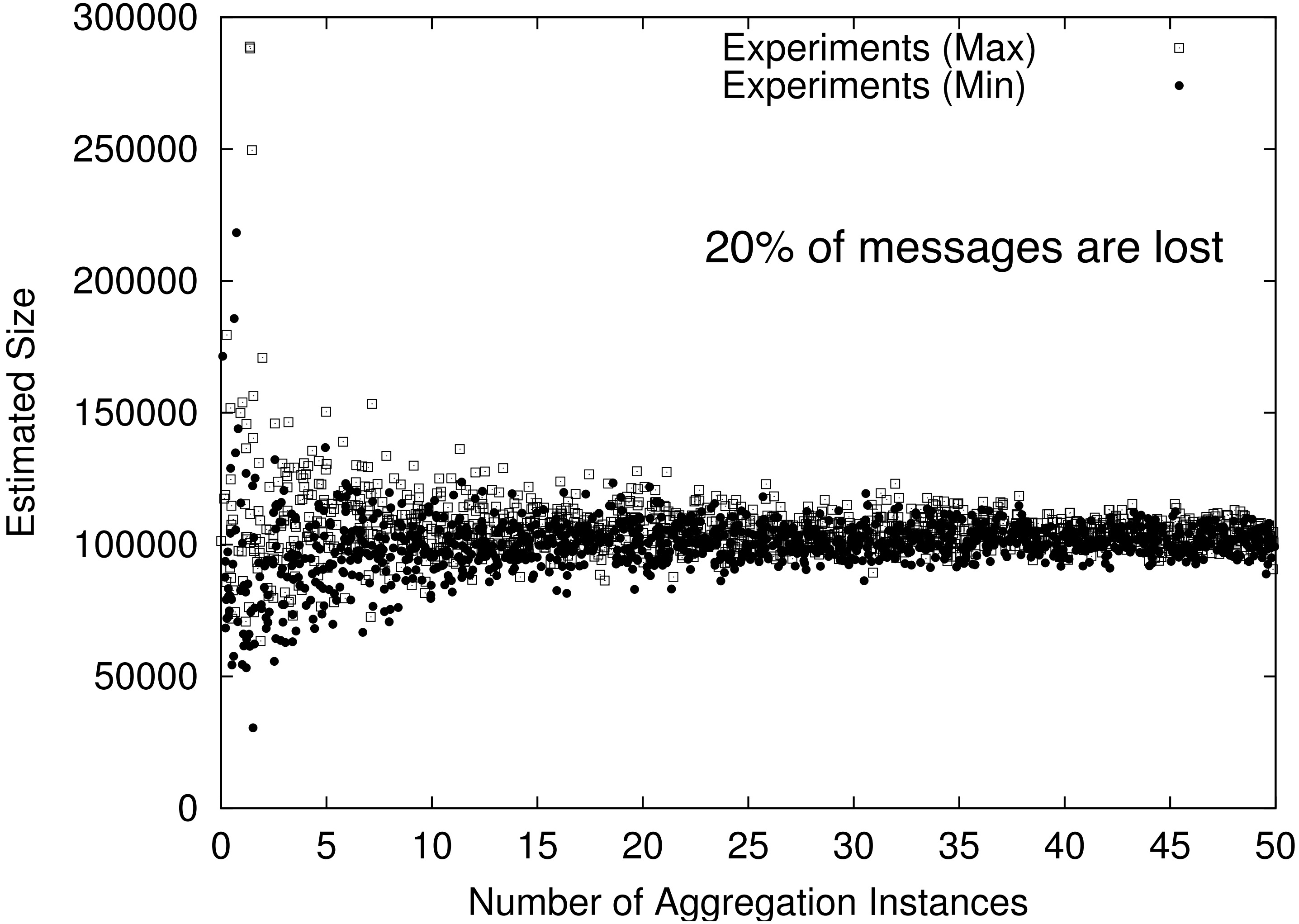
Network size estimation under churn



Network size estimation with multiple aggregation instances



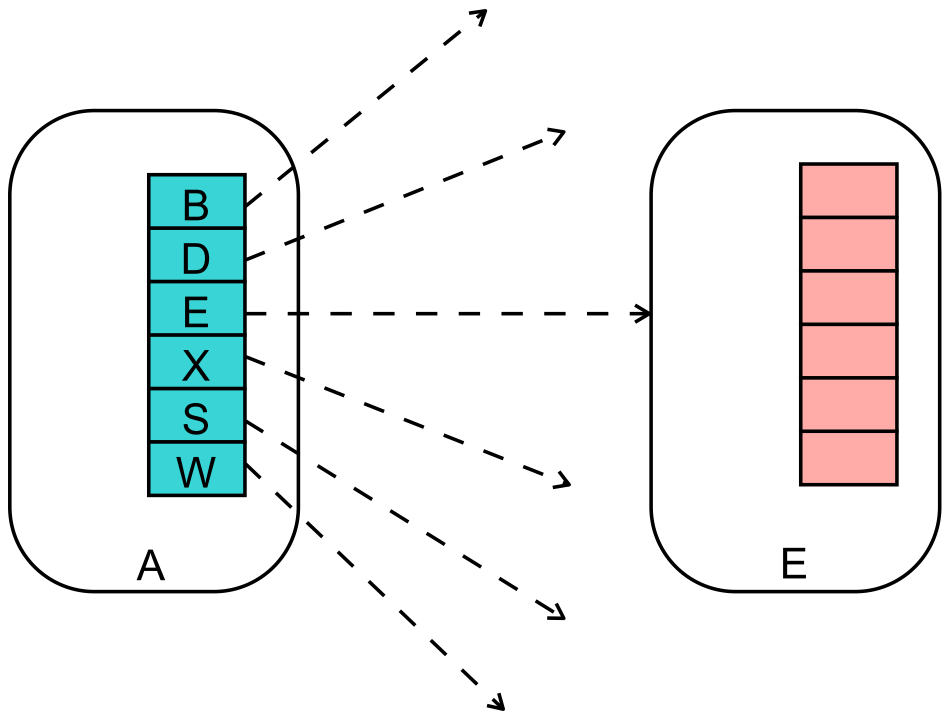
Network size estimation with multiple aggregation instances



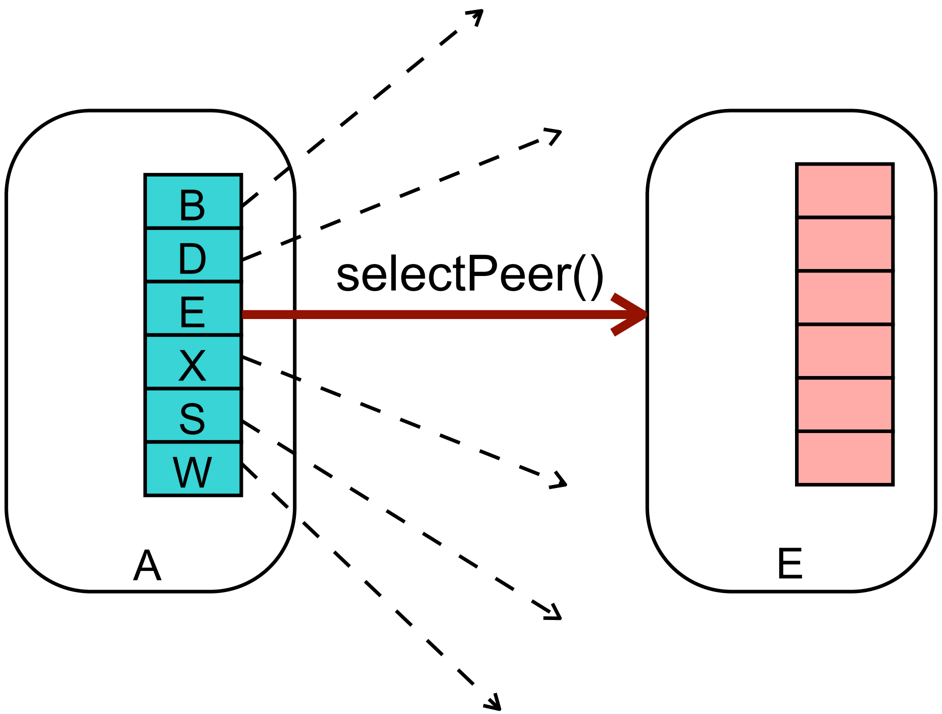
Topology management

- How to ensure that the overlay network topology satisfies certain properties:
  - has a desired structure (connected, random graph, ring, torus, binary tree, etc.)
  - *maintains* the desired structure in a dynamic setting (churn)
- Problem to be solved is *topology management*
- Solution based on epidemics

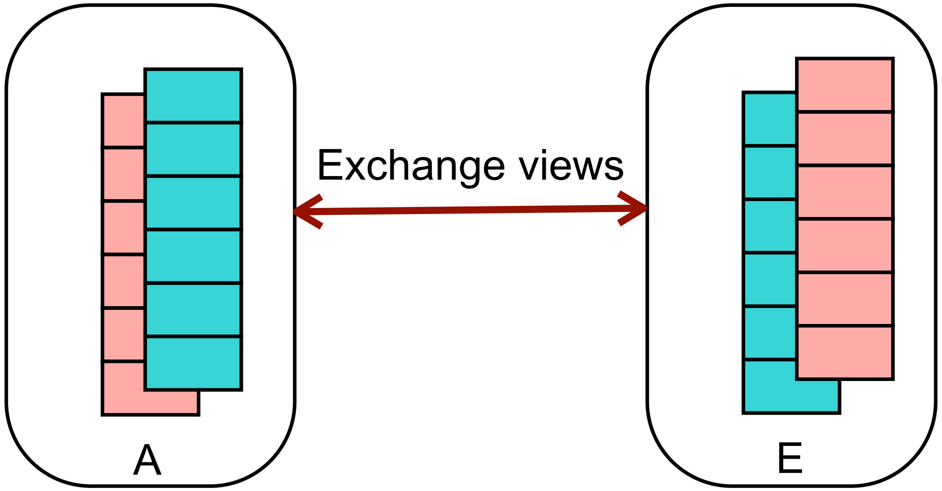
Topology management: example



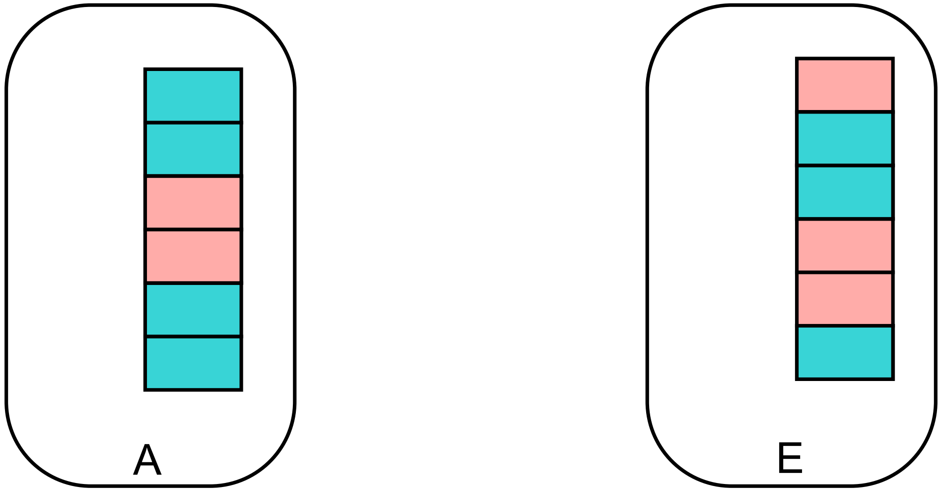
Topology management: example



Topology management: example



Topology management: example

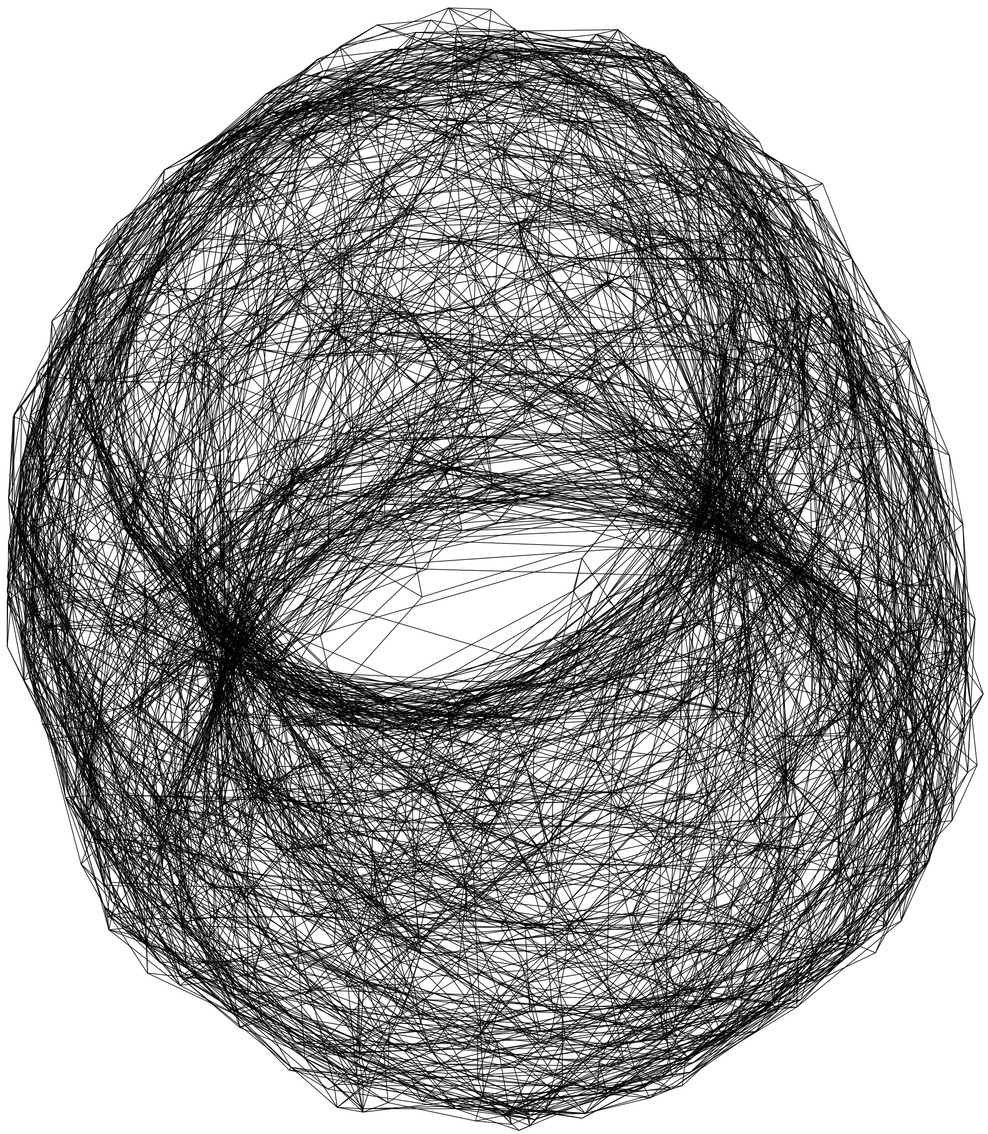


Both peers apply updateState thereby redefining topology

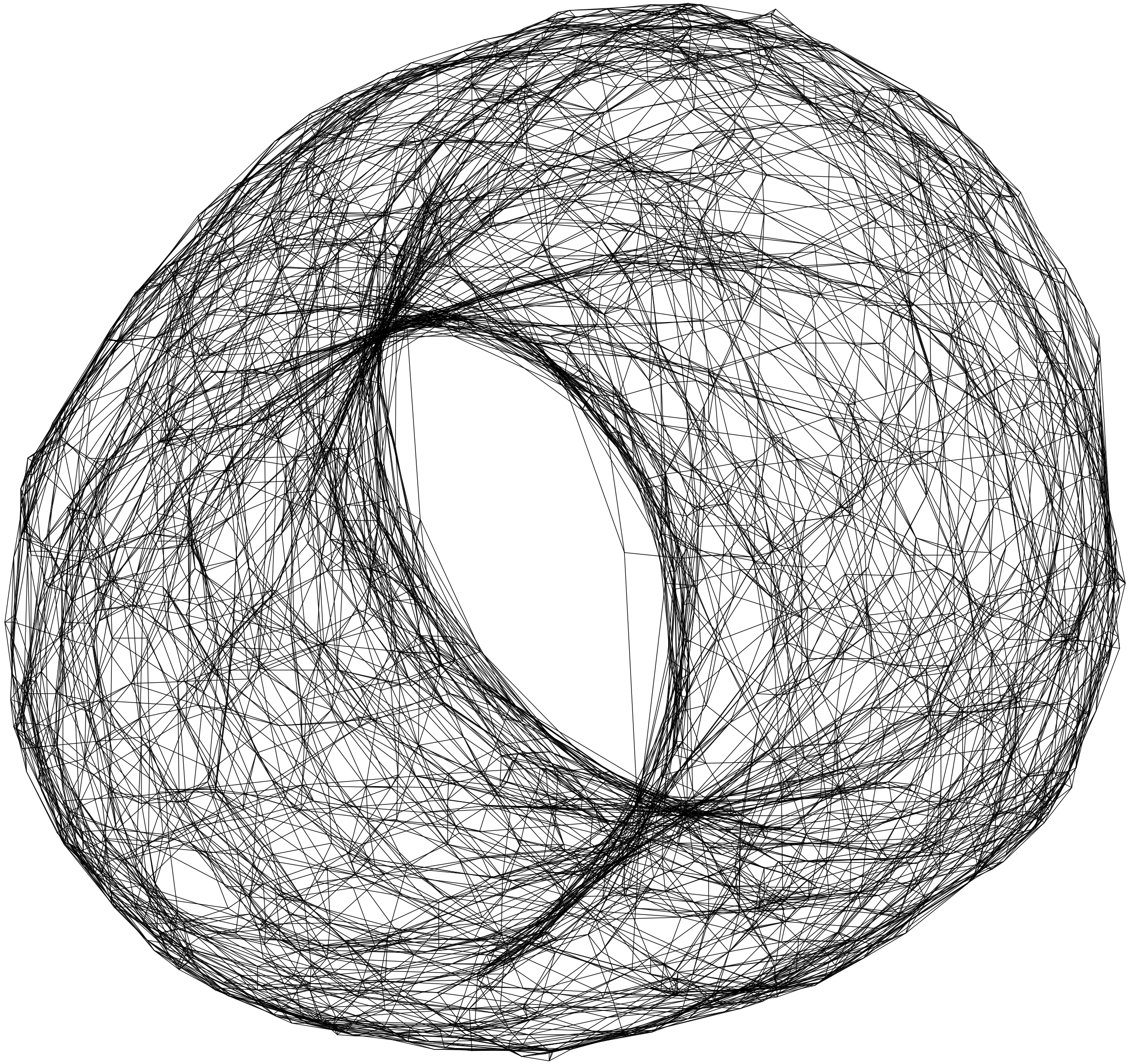
Epidemic-inspired topology management

- *Newscast*
  - Unstructured, almost-random topologies
- *T-man*
  - Wide range of structured topologies including small and large diameter, clustered, sorted, etc.
- Both protocols
  - Extremely robust to node and link failure and node dynamism (churn)
  - Scalable

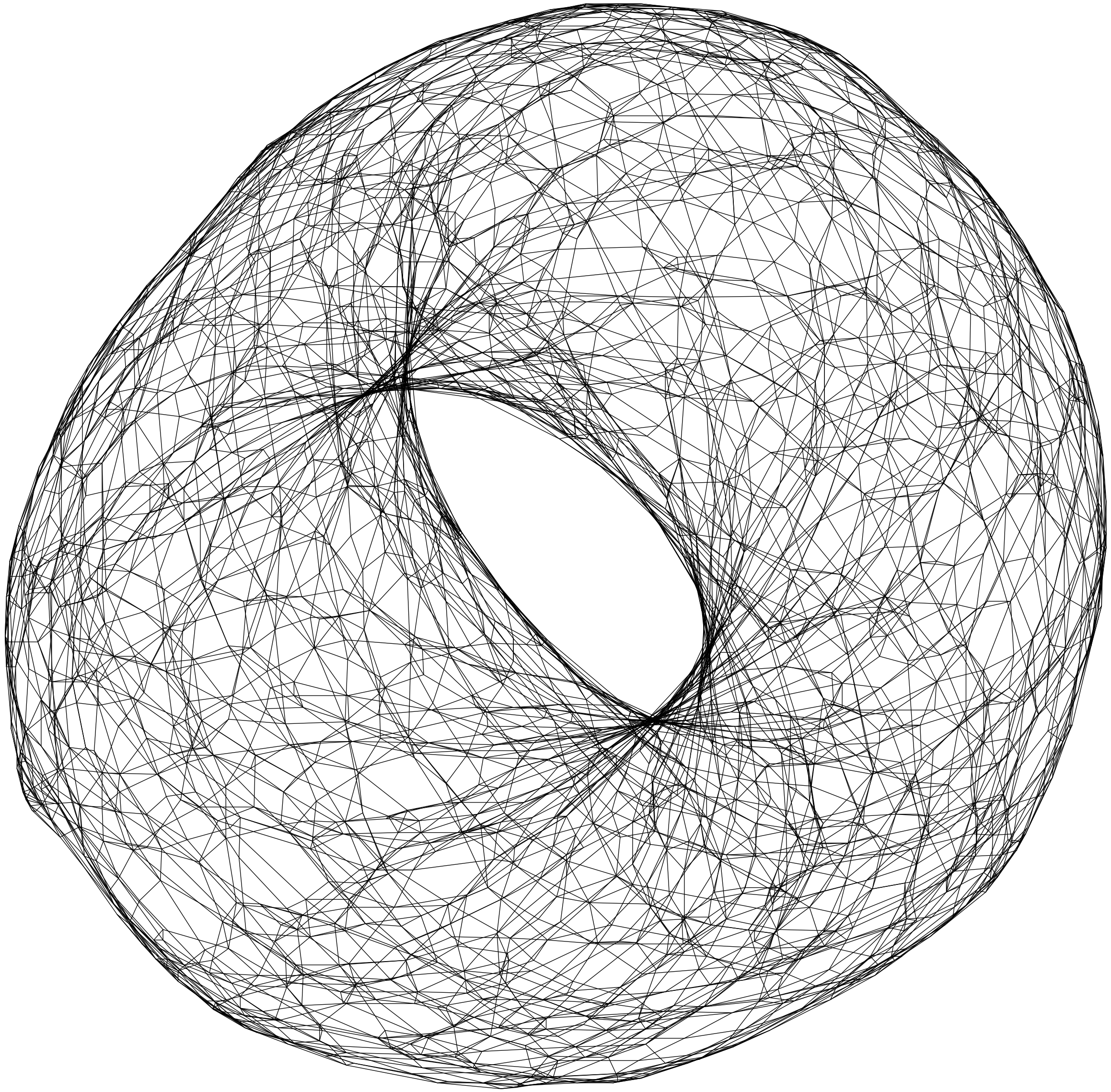
T-Man example: after 3 cycles



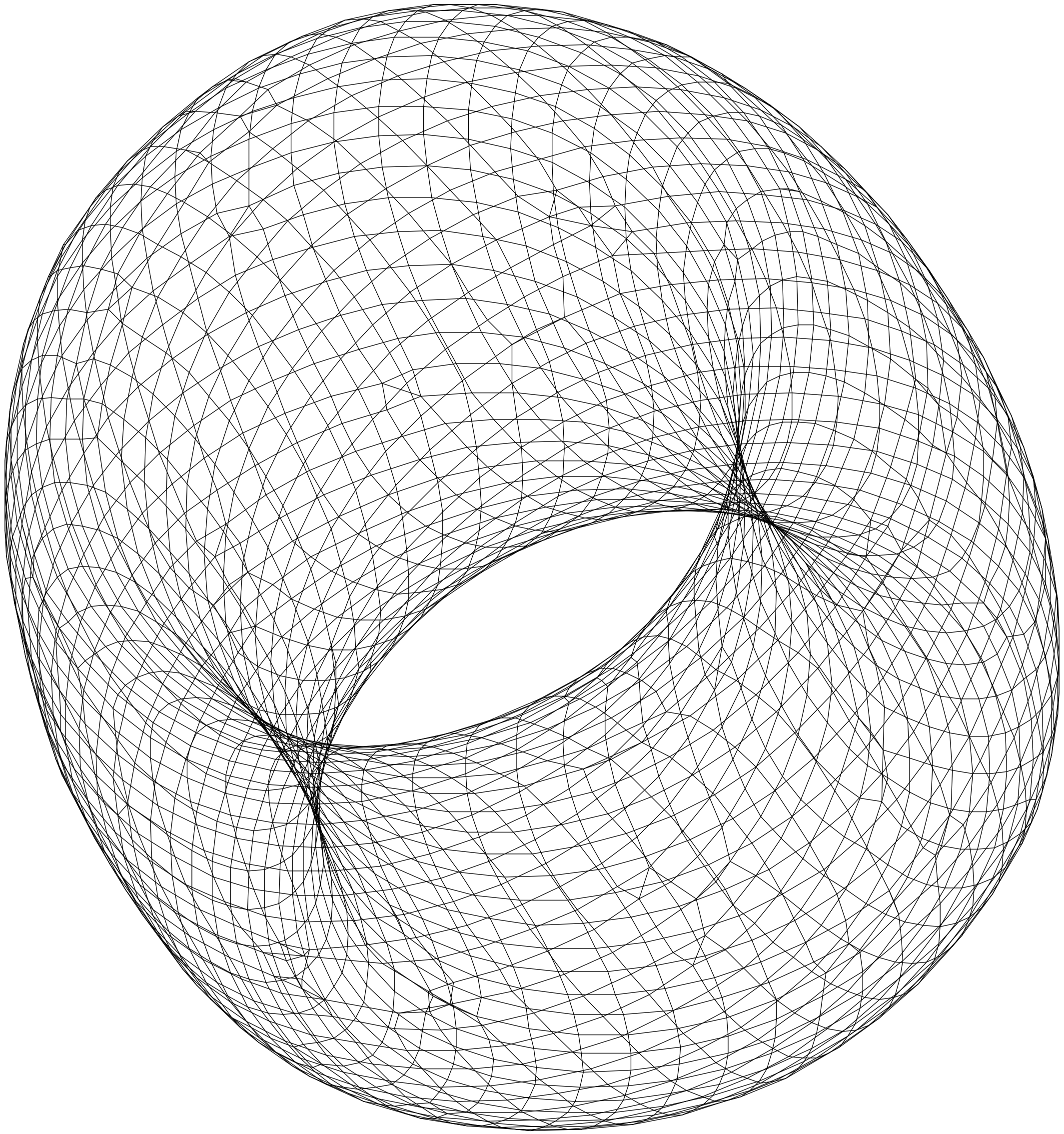
T-Man example: after 5 cycles



T-Man example: after 8 cycles



T-Man example: after 15 cycles



Summary

- Biology is a rich source of inspiration for developing solutions with “nice properties” to technological problems
- To date, we have looked at five biological systems with interesting behavior:
  - Ants: path finding using pheromone, gathering
  - Slime mold amoebae: physical aggregation as a response to collective hunger, using chemotaxis
  - Immune cells: search, recognition, and response to antigens
  - Viruses: epidemic spreading, collective computation